

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
6 December 2001 (06.12.2001)

PCT

(10) International Publication Number  
**WO 01/93250 A1**

(51) International Patent Classification<sup>7</sup>: **G10L 15/22**

(21) International Application Number: **PCT/SE01/01195**

(22) International Filing Date: 29 May 2001 (29.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
09/580,899 30 May 2000 (30.05.2000) US

(71) Applicant (for all designated States except US): **VOXI AB**  
[SE/SE]; P.O. Box 16286, S-103 25 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **LEWIN, Erland**  
[SE/SE]; Bondgatan 81, 4tr, S-116 43 Stockholm (SE).  
**STENIUS, Mårten** [SE/SE]; Furusundsgatan 15, 6tr,  
S-115 37 Stockholm (SE). **ADLER, Daniel** [SE/SE];  
Tideliusgatan 61, S-118 69 Stockholm (SE).

(74) Agents: UDO, Hinz et al.; Stockholms Patentbyrå Zacco AB, P.O. Box 23101, S-104 35 Stockholm (SE).

(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

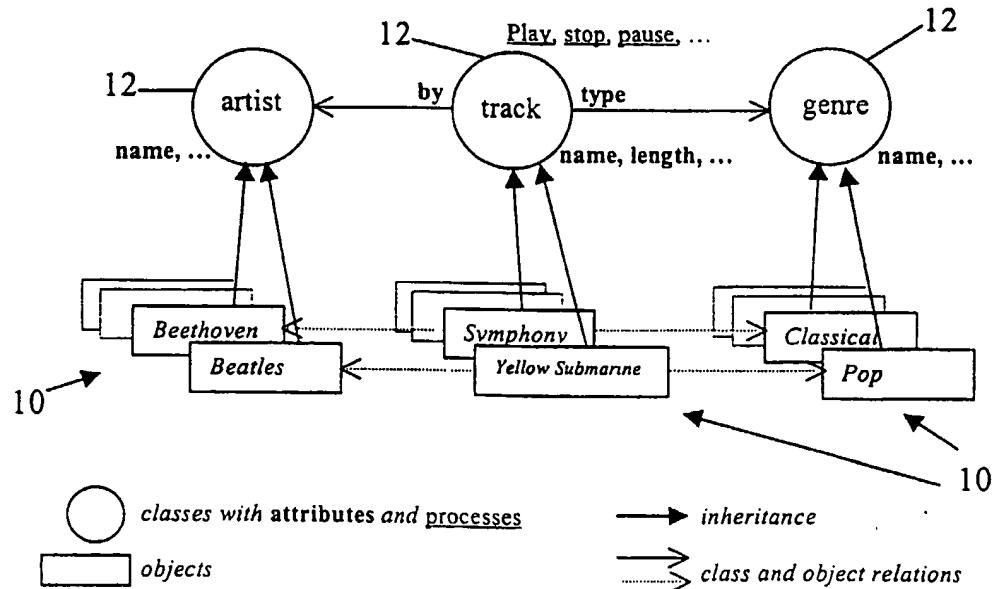
Published:  
— with international search report  
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

[Continued on next page]

(54) Title: LANGUAGE-UNDERSTANDING



**WO 01/93250 A1**



(57) Abstract: The invention relates to a method and a system for separating processing for language-understanding from an application and its functionality. The application is containing functionality within a provided domain, whereby the domain and its functionality e.g. belongs to a speech controlled device such as a car-radio, car-AC, CD-player etc. or almost any program controlled device.

**WO 01/93250 A1**



*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## LANGUAGE-UNDERSTANDING

**Technical Field**

The present invention pertains to a method and a system for separating processing for  
5 language-understanding from an application and its functionality, said application containing  
functionality within a provided domain.

**Background Art**

Conventional speech recognition application programming interfaces (API:s),  
such as Microsoft Speech API™ and Java Speech API™, take input on the form of a  
10 grammar and a lexicon, with little other information on the context or application domain in  
which the language interface is to operate. The output of such API:s is typically a stream of  
words, and an application designer must build a substantial amount of custom code to  
interpret the words and make appropriate application calls.

As illustrated in Fig. 1 of the attached drawings, the conventional speech  
15 recognizer with its API is so to speak glued with custom code to the application itself. The  
custom code provides the “intelligence” in translating a stream of words received from the  
speech recognizer to appropriate application calls. Any translation to actual application  
objects, methods, etc has to be done on a per-case basis in the custom code.

Other speech API:s aim at reducing the amount of custom code, by allowing the  
20 use of modal dialogs. For example, the Philips SpeechMania® 99 product has been  
demonstrated with a pizza ordering application, where a user goes through dialog modes  
involving for instance selecting pizza toppings. A disadvantage of this type of technology is  
that the system will only understand the utterances expected in the given mode. If the user  
25 changes his drink order while the user is expected to select pizza toppings, the system may  
fail to understand this. The degree to which the system ‘understands’ the utterances in this  
kind of interaction is limited; each mode and the utterances valid therein must be anticipated  
by the developers, and directly related to the action the system takes as a response to the user  
input. This also means it requires a substantial amount of interface design work, with  
extensive studies (such as “wizard of oz”-type of settings) to determine every possible phrase  
30 a user might come up with in a given situation.

A widely distributed application of speech recognition and language-  
understanding today is different forms of telephony services. These systems are typically built  
with a central server, which accepts incoming voice calls over standard telephone lines. The  
users are presented with an interactive voice-based interface, and can make choices, navigate

through menus, etc by uttering voice commands. The complete set of software, ranging from the speech recognition, through language-understanding, to application calls, database searches, and audio feedback, resides on the central server. This put high demands on the central server hardware and software, which also must support a large number of simultaneous interactive voice sessions. Typical applications for this type of system is ticket booking, general information services, banking systems, etc. An example of such a system is the "SJ Passenger traffic timetable information system", in use by the Swedish Railway.

Many speech- and language-enabled applications do not use speech recognizer API:s (see description above with respect to the discussion of "conventional speech recognition API:s"). Instead, they implement the whole range of technologies required, from speech recognition through syntactic and semantic (linguistic) processing to the actual application calls and effects. Such designs are called monolithic, since they do not make use of specified API:s to distinguish between different interchangeable modules of the language interaction system, but rather put all components in "one design". An example of such a design is disclosed by, Bertenstam J. et al, "The Waxholm Application Data-Base", Proc. of Eurospeech '95, Vol. 1, pp. 833-836, Madrid, 1995. The "Waxholm system" is a speech-controlled system for search and retrieval of information on boat timetables and services in the Stockholm archipelago. The system implements all relevant linguistic components, such as speech recognition, lexicon, grammar, semantics and application functionality internally.

The field of distributed systems in general deals with the distribution of databases, object repositories, etc over computer networks. The general intent is to provide unified high-level platforms to be used by computer applications that require runtime data to be presented and distributed over a network. One effort to provide a standardized framework for the design of distributed systems is the Common Object Request Broker Architecture (CORBA), proposed by the Object Management Group (OMG). The CORBA architecture is centered around the Object Request Broker (ORB), which handles application (client) calls to a distributed object by providing object stubs (or proxies) on the client-side, on which remote procedure calls are made and transferred to the actual object implementation (server) over the network.

The present invention addresses some fundamental problems that currently arise when language-based interaction is to be performed with multiple application entities present. These can be summarized in three main issues:

1) The lack of a consistent natural language interaction model for different application entities. This means that a multitude of different applications exist with different

and mutually inconsistent linguistic interfaces. The interpretation of the recognized strings of words received from the speech recognizers is done by custom code (see description above with respect to the discussion of “conventional speech recognition API:s”), or even with the complete speech recognition and linguistic processing as an integral part of the application  
5 (see description above with respect to the discussion of “monolithic applications with language-based interaction”), and thus with application-specific solutions. This means that the ways users speak to machines varies and is inconsistent.

2) The lack of transparent interaction using natural language with multiple application entities. Given multiple natural language-enabled applications, there is a lack of  
10 unifying methods to bring the language interfaces together so as to make them accessible at once by the user. Application-specific solutions to distinguish between different sub-functionalities of a system exist (such as prefixing an utterance by “*telephone, ...*” or “*calendar, ...*” to indicate the context of a command), but this is still limited to customized solutions of particular application designs, and the parsing and linguistic processing is still left  
15 to each particular application once the destination of an utterance is determined. Thus, there exists a lack of “*unification of linguistic processing and execution*”, given different accessible applications. As an example of where this type of interaction is problematic, consider a situation when a user wants to control different electronic systems integrated in a car, a stereo and a climate control system. Rather than prefixing each utterance with a destination (by  
20 saying things such as “*radio, louder*”, or “*climate, cooler*”), the system should be able to resolve sentences in the context of both applications simultaneously and understand that the verb “*louder*” is addressed to the radio, and “*cooler*” is addressed to the climate control system, something that currently can only be achieved by building the two applications as one single application unit.

25 3) The requirement to build natural language processing into all entities. Since there are no methods of unifying the linguistic processing of disparate applications in one design (see the two previous points), the full linguistic processing must with conventional techniques be built into each application. This is generally a problem when it comes to efficient resource usage (with respect to memory and processing power, as well as to the  
30 manpower required to develop a working system). Whereas less problematic in centralized design (such as exemplified in the description above with respect to the discussion of “conventional telephony systems”), this problem becomes severe in the case of built-in

systems, portable designs, etc, since such implementations are extremely sensitive to the amount of processing hardware required for a particular application.

#### Summary of the Disclosed Invention

The present invention relates to a method and a system for separating processing for language-understanding from an application and its functionality, said application containing functionality within a provided domain. It intends to solve problems related to prior art and specifically to provide a general means for controlling application means such as a radio, air-condition etc. and other electrically controlled appliances, and software applications on a computer.

In order to achieve the aims of the present invention it sets forth a method of organizing linguistic data describing linguistic interaction with an application specific linguistic logic and a general linguistic understanding logic. The method comprises the steps of:

- separating said application logic from said general logic
- 15        said application logic containing functionality within a predetermined application domain;
- said functionality being provided through a data model;
- reflecting said functionality to said general logic for use in linguistic interaction by providing that said application exports information about words and senses to said general logic; and
- 20        thus providing a distributed consistent linguistic interaction model for different applications using the same general logic to interpret applications with different functionality.

In one embodiment senses are used to associate exported words with objects, attributes and classes that they represent.

25        In another embodiment said information about words comprise objects, attributes, and classes from said object oriented model.

A further embodiment comprises that said objects are nouns, said attributes are adjectives and said classes are verbs.

30        A still further embodiment sets forth that grammars are provided by the application for specific unusual expressions.

Another embodiment of the present invention provides that the general linguistic understanding logic belongs to speech-recognition.

Yet another embodiment provides that the general linguistic understanding logic belongs to text.

In yet another embodiment standard grammar for utterances and phrases in various languages, which are independent of the domain, are built into the general language-understanding linguistic logic.

A further embodiment encompasses that closed word classes and some very common words in each known language are built into the general language-understanding linguistic logic.

5 Further, one embodiment provides that a transfer of words is considered as a two-step process, comprising:

- 10 on-demand establishment of a connection or presence to determine the need of transfer of the application structure to the general language-understanding logic; and
- providing the application-specific linguistic data from the application to the general language-understanding logic.

Another embodiment of the invention comprises that the second step is accomplished by direct transfer, or by providing access through a distributed object system.

15 A still further embodiment comprises that a wireless network is used as interface between said general logic and said application specific logic. In one embodiment the wireless network is operating in accordance with the Bluetooth standard.

The present invention also sets forth a system of organizing linguistic data describing linguistic interaction with an application means for specific linguistic logic and a general 20 linguistic understanding logic engine means comprising:

- separating means to separate said means for specific logic from said engine means, said specific logic means containing functionality within a predetermined application domain;
- said functionality being provided through a data model;
- reflecting means for said functionality to said logic engine means for use in linguistic 25 interaction by providing that said specific logic means exports information about words and senses to said engine means; and
- thus providing a distributed consistent linguistic interaction for different applications using the same general logic engine means to interpret applications with different functionality.

The system according to the present invention is also able to set forth the above 30 method embodiments as disclosed in the attached dependent system claims.

#### Brief Description of the Drawings

The invention is now described in more detail in the form of non-limiting embodiments according to the present invention, clarified with the help of the enclosed drawings, where:

Fig. 1 schematically illustrates conventional speech recognition API:s according to the background art;

Fig. 2 schematically illustrating elimination of “custom code” according to the present invention; and

5 Fig. 3 schematically illustrating an implementing example according to the present invention.

#### Detailed Description of Preferred Embodiments

The present invention provides a method and a system for separating processing for language-understanding from an application and its functionality. This results in a linguistic 10 application programming interface (API) and runtime communication protocol design that allows an application developer to provide natural language interaction, independent of the separation of the interaction logic and the application logic. Examples of such configurations are:

- Separation by a conventional computer network. Such as: A mail-reading application 15 residing on a stationary computer, connected via a computer network to a generic language-understanding engine.
- Integrated equipment. Such as: An integrated music player device, combining the music playing application with the speech interaction logic in one device.
- Separation by a wireless communications channel. Such as: An automobile control 20 system for air conditioning, Hi-Fi equipment etc in the car is connected via a wireless computer network to a generic portable device with basic linguistic capabilities.
- Units present on a dynamically changing network. Such as: An ad-hoc connection 25 established between a personal digital assistant (PDA) and a portable device with speech and language-understanding, using a dynamic wireless network protocol (see for example “Specification of the Bluetooth System – v 1.0B”, Bluetooth Special Interest Group, December 1, 1999).

The key feature of the invention that makes this possible is: The distinction 30 between application-specific logic and general linguistic logic and the separation of these into different system modules. This is described here below with respect to “distinction between application-specific and general linguistic logic”. This is what enables a unified API that can be used in a wide range of settings, such as previously exemplified. Herein below (with respect to “design consequences”) some consequences of this design is described, and below

is also exemplified by describing the implementation of a music player controllable through natural language interaction.

The present invention specifies a manner of organizing linguistic data describing possible linguistic interaction with an application, such that the application may be separate from the language-understanding, and possibly speech recognition aspects. Hence, a distinction between application-specific and general linguistic logic.

The application contains functionality within some given domain (for example an email application, a music player, or a word processor). This functionality can be reflected for use in linguistic interaction by letting the application export words, senses, objects, optional grammars, and optional speech models as detailed below. The object model itself is language-independent, whereas words, senses, grammars, and speech models are language-specific, and by switching these multiple languages can be supported.

The exported words represent objects (nouns), attributes (adjectives), functions (verbs) and possibly other aspects of the application. The information about the words may include their spelling (for textual interaction) and pronunciation (phonetic descriptions for speech interaction), and what language the word is in.

Senses are used to associate the exported words with the objects, attributes and functions that they represent.

The functionality of the application is provided (reflected) as an object model, with the objects representing the data which can be manipulated, methods of the objects being functionality which can be called, and attributes of the objects representing information variables, constants, and relationships between objects.

If the application requires special, unusual ways of expressing things, optionally grammars for this may be provided by the application.

Yet one option is possible organization of information in a speech recognition system, that the application may provide the models required by the speech recognition engine to recognize the words specified by the application. With this organization, the language-understanding engine is not required to have a speech recognition model for the entire language, only the built in words or their phonemes.

The language-understanding engine may contain functionality for speech recognition or text input using a speech model, parsing of a recognized string of words, and language-understanding using a grammar and a set of words. The speech model being used is a combination of a standard speech model (see below) residing in the language-understanding engine, and an optional application-specific speech model provided by the application. The

grammar is a combination of a standard grammar (see below) residing in the language-understanding engine, and an optional application-specific grammar provided by the application. The set of words is a combination of a set of standard words (see below) residing in the language-understanding engine, and an application-specific set of words provided by the application. Given this, the speech recognition, parsing and language-understanding is done in the context of the objects and senses exported by the application, to resolve the actual semantic meaning of an utterance, and thereafter execute the resolved action (method) with appropriate parameters by calling the application back through the invention API. The data contained in the language-understanding engine is language-specific, and multiple languages may be supported simultaneously. The language-understanding engine has the following information:

- Standard Grammars: The standard grammar for utterances and phrases in various languages, which are independent of the domain may be built into the language-understanding engine.
- 15 • Standard Words: The so-called “closed” word classes (such as pronouns, prepositions, conjunctions, articles etc) and some very common words in each known language may be built into the language-understanding engine.
- 20 • Standard Speech Models: If the language-understanding unit also does speech recognition, it may contain speech recognition models (such as Hidden Markov Model statistical speech models) for the languages that it knows.

Moreover, In the following is discussed some consequences of the solution in closer detail, with respect to design consequences:

One consequence is with respect to elimination of “custom code”. Reference is made to Fig. 2 of the attached drawings, where it is described that API according to the present invention allows applications to describe their functionality in a well-defined way, thus avoiding the problems arising from disparate custom solutions.

Another consequence of the invention design is that the “application-specific data must be transferred to the language-understanding unit” to enable it to correctly process utterances in the context of that application. This transfer can be considered as a two-step process:

1. The on-demand establishment of a connection or presence to determine the need of transfer of the application structure to the language-understanding engine.
2. Providing the application-specific linguistic data from the application to the language-understanding engine. This can be done by direct transfer, or by providing access through a distributed object system.

5 A data model consists of objects with attributes. Relations between objects are expressed as attributes. The data model may also contain methods, which may be related to the objects. In some data models, all objects are divided up into classes, which are represented separately. The interpretations of a class is that objects are in some sense uniform instances of  
10 a class, often sharing the same attributes. This is a specialization of what is meant by a data model. There are often primitive objects, such as integers and text, which are not actually represented as objects, but treated as a special case. This is also a specialisation of what is meant by a model. In some models the methods are not related to objects. Examples of data structures included in this definition of a data model are relational databases, object-oriented  
15 models, and structures in programming languages such as C.

Yet one consequence is that by the separation of application-specific data and the logic of the generic language-understanding engine, the mechanism is provided for both a "consistent natural language interaction model", and a "transparent (unified) natural language interaction" when multiple applications are accessible by the user. For instance, assuming that  
20 a user is in possession of a generic language-understanding unit, and gets access to more than one application using the present solution to present its functionality. Then, using the connection medium in place (a computer network, an ad-hoc wireless network, inter-process communication within one host, etc), each application exports its set of words, objects, methods, etc to the language-understanding unit. The language-understanding unit then fills in  
25 its standard (generic) grammar structures with these data models, and the user can interact with one single linguistic description that is capable of controlling each application directly. This shows that points 1 and 2 of the problem description are properly addressed by the solution according to the present invention. Furthermore, since the speech recognition and language-understanding logic resides in the generic language-understanding unit, this logic  
30 will not be needed in the application units, which also shows that point 3 of the problem description is properly addressed.

Yet another consequence is that techniques for building distributed object repositories exist (as exemplified above with respect to discussion of "distributed systems" according to background art), and it is possible to use similar methods to implement the communication

between the application and the linguistic engine. Method calls for objects in a distributed situation could for instance be implemented by RPC-style (remote procedure call) mechanisms. The exact nature of the implementation will of course depend on the medium of transport that separates the application and the linguistic engine, which for instance may 5 be separated on different hosts on Internet, different units on a wireless network or located on the same computer and simply separated by local inter-process communication or just native application calls.

Fig. 1 illustrates, with reference to the prior art, that a current Speech API:s requires a substantial amount of custom code to build an interactive linguistic interface to an 10 application. The conventional speech recognizer with its API is so to speak glued with custom code to the application itself. The custom code provides the “intelligence” in translating a stream of words received from the speech recognizer to appropriate application calls. Any translation to actual application objects, methods, etc has to be done on a per-case basis in the custom code.

15 Fig. 2 shows the API according to the present invention allowing applications to describe their functionality in a well-defined way, thus avoiding the problems arising from disparate custom solutions. The present invention effectively eliminates large parts of the “custom code” required for conventional speech API:s (see description above with respect to background art). According to the present invention, the API is moving as seen by an 20 application programmer “one level up”, as evident when comparing Fig. 1 to Fig. 2.

As illustrated in Fig. 3, a practical use of the present invention is demonstrated. Described is the implementation of a music player controlled by speech and natural language, and controllable over a computer network, using the present invention as an implementation platform. The music player makes a set of words, senses, and objects representing the 25 application functionality available to the natural language-understanding engine. Assume the following basic functionality:

- Support for basic “music player” commands: play, stop, restart.
- Support for songs, artists and genres.

To model this simple application, an object-oriented design, for example, through such 30 software as Java or C++ etc., is created, as shown in Fig. 3. Given this, the application exports the following data, in accordance with the description above relating to the discussion of “distinction between application-specific and general linguistic logic”:

“Words”. The music player exports the following words, for example through a wireless interface to a language engine in accordance with the present invention, with associated textual and phonetic information:

- *Nouns* for the available artists (such as “Beethoven”, or “Beatles”).
- 5 • *Nouns* (or combined “pseudo-nouns”) for the available songs titles (such as “Symphony” or “Yellow Submarine”).
- *Nouns* for general use in the context of a music player, such as “track”, “song”, and “music”.
- *Adjectives* for the available music genres (such as “classical” or “rock”).
- 10 • Verbs for the main functions to execute (such as “play”, “stop” and “continue”).

Depicted arrows in Fig. 3 illustrate inheritance (filled in arrow), thinner arrow (classes), dotted arrows (objects).

“Senses”. The application exports the following senses, to connect each word to appropriate objects 10, attributes, and functions: Each artist noun is connected to the appropriate object representing the artist. The noun “Beethoven” is connected to the object representing Beethoven, and so on. Similarly, each song title is connected to an object representing that song. Each general noun is connected to a corresponding object class. For instance, the nouns “track” and “music” can be connected to the song class. Each adjective is connected to an object representing the genre. The adjective “classical” is connected to the object representing classical music, and so on. Each verb is connected to an appropriate object process, to represent the action to be taken when the verb is uttered. For instance, the verb “play” is connected to a playback process of the object class representing songs.

“Objects”. An object model with classes 12 and instances, for example, in an object oriented programming language, is built to reflect the music application data and functionality. Object classes 12 are created for songs, artists and genres. Processes for the desired functionality are assigned to these classes 12. For instance, a playback process is associated to the song class, and so on. Objects are instantiated for these classes 12, to represent the actual artists, songs, and genres in the system: A Beethoven instance of the artist class, a “Yellow Submarine” instance of the song class, a “classical music” instance of the genre class, and so on. Relevant attributes are set of the instance objects 10, to represent different properties of the underlying data. This can be to reflect the duration of

a track, by a simple integer, or the genre of a song, which is represented by an object association, such as associating the “Symphony” object to the “classical music” object.

“Optional grammars and speech models”. In this case, no special grammar rules need to be applied, the basic form of the allowed sentences residing in the language-understanding engine is sufficient. Similarly, no special speech models are required in this case, the phonetic descriptions of the words exported is sufficient to act as input to the standard speech model residing in the language-understanding engine.

The language-understanding engine contains an application-independent grammar description, along with a set of standard words (prepositions, etc), and generic speech models. The grammar description contains entries, such as: <verb> <noun>; or <verb> <preposition> <adjective> <noun>; or more intricate linguistic definitions. These general grammar descriptions, in the context of the data exported by the music application, result in a complete mapping between possible utterances and a mapping of these to appropriate actions to execute by making calls to the exported application interface.

15 As a runtime functionality example, assuming the following utterance from a user:

“Play a classical track”. This is recognized by the standard speech recognizer, which produces a string of words. The string of words is fed into the language-understanding mechanism of the engine, and resolved in the context of the existing grammar, objects 10, classes 12, using the method described in the previous application incorporated above by reference filed by the same assignee, Voxi AB, as the present invention. Very briefly, this resolution manages to find a song object associated to the class representing “classical music”. This also involves filtering out classes 12 that have a method associated to the word “play”, and so on. Once this resolution is finished, the playback method is called on the resolved music application object. 25 In this example, an outcome could be that the playback method of the object representing “Symphony” is called, resulting in a classical track being played to the user.

It is appreciated that means and logic mentioned throughout the above description could be realized through software, hardware, or by a combination of both as known in the art.

30 The present invention has been described with non-limiting examples and embodiments. It is the attached set of claims that describe all possible embodiments for a person skilled in the art.

**Claims**

1. A method of organizing linguistic data describing linguistic interaction with an application specific linguistic logic and a general linguistic understanding logic comprising:
  - separating said application logic from said general logic
  - 5 said application logic containing functionality within a predetermined application domain;
    - said functionality being provided through a data model;
    - reflecting said functionality to said general logic for use in linguistic interaction by providing that said application exports information about words and senses to said general
    - 10 logic; and
    - thus providing a distributed consistent linguistic interaction model for different applications using the same general logic to interpret applications with different functionality.
2. A method according to claim 1, wherein senses are used to associate exported words with objects, attributes and classes that they represent.
- 15 3. A method according to claims 1 or 2, wherein said information about words comprise objects, attributes, and classes from said data model.
4. A method according to claims 1-3, wherein said objects are nouns, said attributes are adjectives and said classes are verbs.
5. A method according to claims 1-4, wherein grammars are provided by the
- 20 application for specific unusual expressions.
6. A method according to claims 1-5, wherein the general linguistic understanding logic belongs to speech-recognition.
7. A method according to claim 6 wherein the application provides the models required by the speech recognition logic to recognize the words specified by the application
- 25 8. A method according to claims 1-5, wherein the general linguistic understanding logic belongs to text.
9. A method according to claims 1-8, wherein standard grammar for utterances and phrases in various languages, which are independent of the domain, are built into the general language-understanding linguistic logic.
- 30 10. A method according to claims 1-9, wherein closed word classes and some very common words in each known language are built into the general language-understanding linguistic logic.
11. A method according to claims 1-10, wherein a transfer of words is considered as a two-step process, comprising:

on-demand establishment of a connection or presence to determine the need of transfer of the application structure to the general language-understanding logic; and

providing the application-specific linguistic data from the application to the general language-understanding logic.

5 12. A method according to claim 11, wherein the second step is accomplished by direct transfer, or by providing access through a distributed object system.

13. A method according to claims 1-12, wherein a wireless network is used as interface between said general logic and said application specific logic.

10 14. A method according to claim 13, wherein the wireless network is operating in accordance with the Bluetooth standard.

15 15. A system of organizing linguistic data describing linguistic interaction with a application means for specific linguistic logic and a general linguistic understanding logic engine means comprising:

separating means to separate said means for specific logic from said engine means,  
15 said specific logic means containing functionality within a predetermined application domain;  
said functionality being provided through a data model;

reflecting means for said functionality to said logic engine means for use in linguistic interaction by providing that said specific logic means exports information about words and senses to said engine means; and

20 thus providing a distributed consistent linguistic interaction for different applications using the same general logic engine means to interpret applications with different functionality.

16. A system according to claim 15, wherein senses are used to associate exported words with objects, attributes and classes that they represent.

17. A system according to claims 15 or 16, wherein said information about words 25 comprise objects, attributes, and classes from said data model.

18. A system according to claims 15-17, wherein said objects are nouns, said attributes are adjectives and said classes are verbs.

19. A system according to claims 14-18, wherein grammars are provided by the application for specific unusual expressions.

30 20. A system according to claims 15-19, wherein the general linguistic understanding logic belongs to speech-recognition.

21. A system according to claim 20, wherein the application provides the models required by the speech recognition logic to recognize the words specified by the application.

22. A system according to claims 15-19, wherein the general linguistic understanding logic belongs to text.

23. A system according to claims 15-22, wherein standard grammar for utterances and phrases in various languages, which are independent of the domain, are built into the general language-understanding linguistic logic.

5 24. A system according to claims 15-23, wherein closed word classes and some very common words in each known language are built into the general language-understanding linguistic logic.

10 25. A system according to claims 15-24, wherein a transfer of words is considered as a two-step process, comprising:

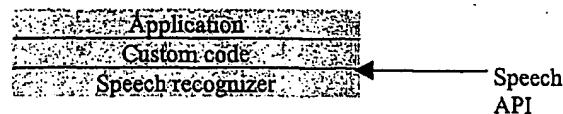
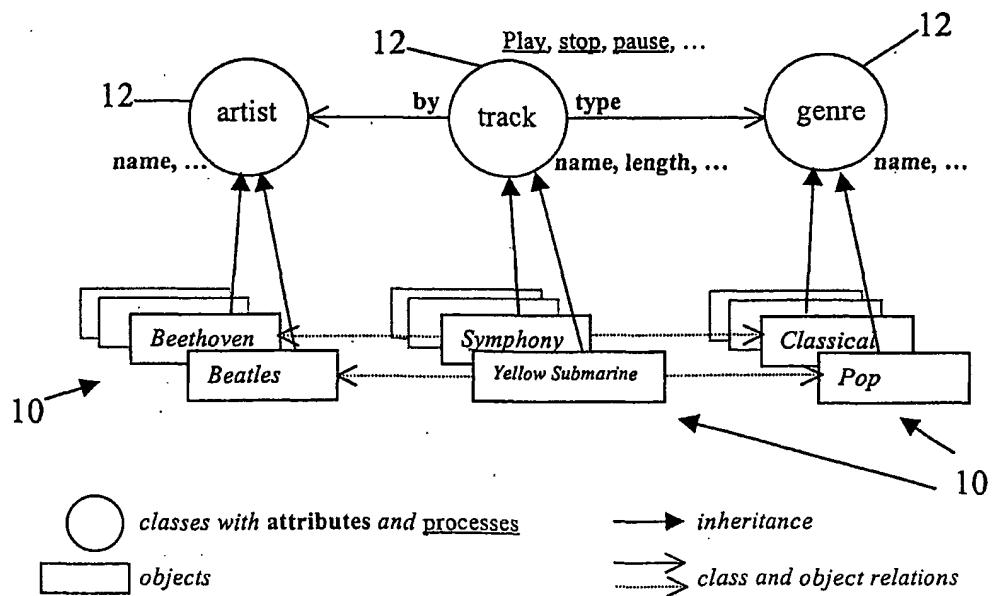
on-demand establishment of a connection or presence to determine the need of transfer of the application structure to the general language-understanding logic; and

providing the application-specific linguistic data from the application to the general language-understanding logic.

15 26. A system according to claim 25, wherein the second step is accomplished by direct transfer, or by providing access through a distributed object system.

27. A system according to claims 15-26, wherein a wireless network is used as interface between said general logic and said application specific logic.

20 28. A system according to claim 27, wherein the wireless network is operating in accordance with the Bluetooth standard.

**FIG. 1****FIG. 2****FIG. 3**

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/SE 01/01195

## A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G10L 15/22

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G10L, B60R

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE, DK, FI, NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO INTERNAL, WPI DATA, INSPEC, COMPENDIX, PAJ

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 0021074 A1 (LERNOUT & HAUSPIE SPEECH PRODUCTS N.V.), 13 April 2000 (13.04.00), page 1, line 31 - page 9, line 6, claims 1,3, abstract --	1-28
A	1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 21-24 April 1997, Stahl H. et al: "Controlling limited-domain applications by probabilistic semantic decoding of natural speech", pages 1163-1166 vol. 2; figure 1, paragraph 5 --	1-28
A	US 5682542 A (H. ENOMOTO ET AL.), 28 October 1997 (28.10.97), abstract --	1-28

 Further documents are listed in the continuation of Box C. See patent family annex.

\* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "B" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

28 August 2001

Date of mailing of the international search report

25-09-2001

Name and mailing address of the ISA/  
Swedish Patent Office  
Box 5055, S-102 42 STOCKHOLM  
Facsimile No. +46 8 666 02 86

Authorized officer

Asa Häggren/AE  
Telephone No. +46 8 782 25 00

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/SE 01/01195

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Interactive Voice Technology for Telecommunications Applications, 1998. IVTTA'98. Proceedings. 1998 IEEE 4th Workshop, 29-30 Sept. 1998, Hanrieder G. et al: "Integration of a mixed-initiative dialogue manager into commercial IVR platforms", pages 77-82; see figure 1 and page 79, right hand column, paragraph C  -- -----	1-28

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

PCT/SE 01/01195

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO	0021074 A1	13/04/00	AU	6225199 A	26/04/00
US	5682542 A	28/10/97	CA	2089842 A,C	22/08/93
			DE	69328452 D,T	07/09/00
			EP	0557205 A,B	25/08/93
			JP	2801459 B	21/09/98
			JP	5233690 A	10/09/93

**RIGHT AVAILABLE COPY**